**Amendments to the Claims**

Please amend claims 1, 15, 26, 33 and add new claims 37 and 38 as indicated below. All claims are listed below, with currently amended claims so marked. This listing of claims will replace all prior versions, and listings, of claims in the application:

1. (Currently Amended) A method of processing a video stream, comprising:

(a) detecting a request to randomly access a particular frame;

(b) maintaining a list of frame dependencies identifying at least a set of frames required to decode the particular frame; and

(c) determining based at least in part on the list of frame dependencies whether a decoded version of the particular frame is in a decoded frame cache, said cache configured to store an arbitrary number of previously decoded frames, and if it is not and if the particular frame has a frame dependency:

(i) determining a frame dependency for the particular frame;

(ii) determining which of the frames in the frame dependency are in the decoded frame cache;

(iii) decoding any frame in the frame dependency that is not in the decoded frame cache and placing it in the decoded frame cache; and

(iv) using at least one of the decoded frames in the frame dependency to decode the particular frame to create a decoded version of the particular frame.

2. (Previously Presented) The method of claim 1, wherein the request to playback a particular frame is part of a request to perform frame-by-frame backward playback and part (c) is performed for successively earlier frames with respect to the particular frame as part of the frame-by-frame backward playback.

3. (Original) The method of claim 1, wherein part (i) is performed whether or not it is determined that a decoded version of a particular frame is in the decoded frame cache without part (iv) being performed.

4. (Original) The method of claim 1, wherein the particular frame may be an I, P, or B frame of MPEG compressed video.

5. (Original) The method of claim 1, wherein the frame dependency is an immediate frame dependency.

6. (Previously Presented) The method of claim 5, wherein the at least some of the decoded frames referred to in part (iv) are those frames in the immediate dependency.

7. (Previously Presented) The method of claim 5, wherein part (c) includes recursion where frames in the immediate frame dependency of the frame of interest are not in the decoded frame cache.

8. (Previously Presented) The method of claim 1, wherein part (c) includes a loop with a terminating condition that all frames on which the particular frame is dependent have been decoded.

9. (Original) The method of claim 1, wherein decoded frames are replaced in the decoded frame cache according to a least recently used policy.

10. (Original) The method of claim 1, wherein an index is used to represent each frame in the frame dependency.

11. (Original) The method of claim 1, wherein the frame dependency is determined through a look-up table.

12. (Original) The method of claim 11, wherein the frame dependency is determined through successive uses of a look-up table.

13. (Original) The method of claim 1, wherein the decoded frame cache includes a data structure.

14. (Original) The method of claim 1, wherein the decoded frame cache includes a section of main memory.

15. (Currently Amended) An article comprising:

a computer readable medium having instructions thereon which when executed cause a computer to:

(a) detect a request to randomly access a particular frame; and

(b) maintaining a list of frame dependencies identifying at least a set of frames required to decode the particular frame;

(c) determine base at least in part on the list of frame dependencies whether a decoded version of the particular frame is in a decoded frame cache, said cache configured to store an arbitrary number of previously decoded frames, and if it is not and if the particular frame has a frame dependency:

(i) determine a frame dependency for the particular frame;

(ii) determine which of the frames in the frame dependency are in the decoded frame cache;

(iii) decode any frame in the frame dependency that is not in the decoded frame cache and place it in the decoded frame cache; and

(iv) use at least and of the decoded frames in the frame dependency to decode the particular frame to create a decoded version of the particular frame.

16. (Previously Presented) The article of claim 15, wherein the request to playback a particular frame is part of a request to perform frame-by-frame backward

playback and part (c) is performed for successively earlier frames with respect to the particular frame as part of the frame-by-frame backward playback.

17. (Original) The article of claim 15, wherein part (i) is performed whether or not it is determined that a decoded version of a particular frame is in the decoded frame cache without part (iv) being performed.

18. (Original) The article of claim 15, wherein the frame dependency is an immediate frame dependency.

19. (Previously Presented) The article of claim 18, wherein the at least some of the decoded frames referred to in part (iv) are those frames in the immediate dependency.

20. (Previously Presented) The article of claim 18, wherein part (c) includes recursion where frames in the immediate frame dependency of the frame of interest are not in the decoded frame cache.

21. (Previously Presented) The article of claim 15, wherein part (c) includes a loop with a terminating condition that all frames on which the particular frame is dependent have been decoded.

22. (Original) The article of claim 15, wherein decoded frames are replaced in the decoded frame cache according to a least recently used policy.

23. (Original) The article of claim 15, wherein an index is used to represent each frame in the frame dependency.

24. (Original) The article of claim 15, wherein the frame dependency is determined through a look-up table.

25. (Original) The article of claim 24, wherein the frame dependency is determined through successive uses of a look-up table.

26. (Currently Amended) A computer system including:

a processor and video processing circuitry;

a display; and

memory including instructions which when executed cause the processor and video processing circuitry to:

(a) detect a request to randomly access a particular frame; and

E1 (b) maintain a list of frame dependencies identifying at least a set of frames required to decode the particular frame;

(c) determine whether a decoded version of the particular frame is in a decoded frame cache, said cache configured to store an arbitrary number of previously decoded frames, and if it is not and if the particular frame has a frame dependency:

(i) determine a frame dependency for the particular frame;

(ii) determine which of the frames in the frame dependency are in the decoded frame cache;

(iii) decode any frame in the frame dependency that is not in the decoded frame cache and place it in the decoded frame cache; and

(iv) use at least and of the decoded frames in the frame dependency to decode the particular frame to create a decoded version of the particular frame.

(d) provide the decoded version of the particular frame for displaying on the display.

27. (Previously Presented) A method for randomly accessing a first frame of a video stream, comprising:

maintaining a list of frame dependencies identifying at least a set of frames required to decode the first frame;

determining a decoding of the first frame is not in a decoded frame cache;

determining, based at least in par on the list of frame dependencies, a first frame dependency for the first frame comprising frames required to decode the first frame;

decoding at least one of the frames of the frame dependency not present in the decoded frame cache, and placing it in the decoded frame cache; and

decoding the first frame using at least one of the decoded frames in the decoded frame cache.

28. (Original) The method of claim 27, further comprising:

decoding each frame of the frame dependency not present in the decoded frame cache, and placing them in the decoded frame cache.

29. (Original) The method of claim 27, further comprising:

recursively decoding the second frame of the frame dependency.

30. (Original) A method according to claim 27 for reverse playback of frames of the video stream, comprising:

determining a second frame is not in the decoded frame cache, the second frame following the first frame in the video stream;

determining a second frame dependency for the second frame comprising frames required to decode the second frame;

decoding at least one of the frames of the frame dependency not present in the decoded frame cache, and placing it in the decoded frame cache; and

decoding the second frame using at least one of the decoded frames in the decoded frame cache.

31. (Original) The method of claim 30, further comprising:

playing the second frame and then the first frame.

32. (Original) The method of claim 30, wherein the second frame is an immediately following frame of the first frame.

33. (Currently Amended) An article comprising a machine-accessible media having associated data for randomly accessing a first frame of a video stream, wherein the data, when accessed, results in a machine performing:

maintaining a list of frame dependencies identifying at least a set of frames required to decode the first frame;

determining a decoding of the first frame is not in a decoded frame cache, said cache configured to store an arbitrary number of previously decoded frames;

determining, based at least in par on the list of frame dependencies, a first frame dependency for the first frame comprising frames required to decode the first frame;

decoding at least one of the frames of the frame dependency not present in the decoded frame cache, and placing it in the decoded frame cache; and

decoding the first frame using at least one of the decoded frames in the decoded frame cache.

34. (Original) The article of claim 33 wherein the machine-accessible media further includes data, when accessed, results in the machine performing:

decoding each frame of the frame dependency not present in the decoded frame cache, and placing them in the decoded frame cache.

35. (Original) The article of claim 33 wherein the machine-accessible media further includes data, when accessed, results in the machine performing:

recursively decoding the second frame of the frame dependency.

36. (Original) The article of claim 33 wherein the machine-accessible media further includes data for reverse playback of frames of the video stream, when accessed, results in the machine performing:

determining a second frame is not in the decoded frame cache, the second frame following the first frame in the video stream;

determining a second frame dependency for the second frame comprising frames required to decode the second frame;

decoding at least one of the frames of the frame dependency not present in the decoded frame cache, and placing it in the decoded frame cache; and

decoding the second frame using at least one of the decoded frames in the decoded frame cache.

37. (New) A method of caching decoded frames of a video in a decoded frame cache configured to store an arbitrary number of previously decoded frames, comprising:

maintaining a list of frame dependencies identifying at least a set of frames required to decode a particular frame of the video;

determining based at least in part on the list of frame dependencies that a decoded version of the particular frame is not in the decoded frame cache; and

determining if the particular frame has a frame dependency, and if so:

determining a frame dependency for the particular frame,

determining which of the frames in the frame dependency are in the decoded frame cache,

decoding any frame in the frame dependency that is not in the decoded frame cache and placing it in the decoded frame cache, and

using at least one of the decoded frames in the frame dependency to decode the particular frame to create a decoded version of the particular frame.

38. (New) The method of claim 37, further comprising:

detecting a request to randomly access the particular frame;

wherein the request to playback the particular frame is part of a request to perform frame-by-frame backward playback.